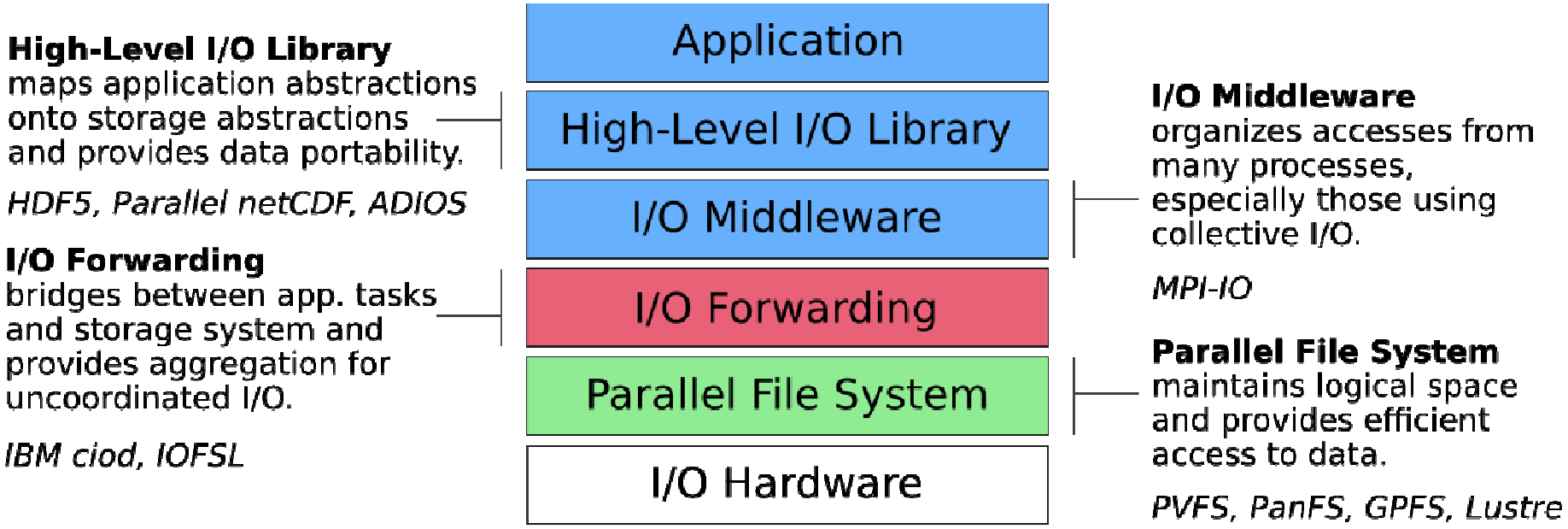
# Next Generation I/O Panel
# HEC-FSIO 2011

Dries Kimpe <dkimpe@mcs.anl.gov>
Argonne National Laboratory

# Current I/O Software Stack




(Figure R. Ross)

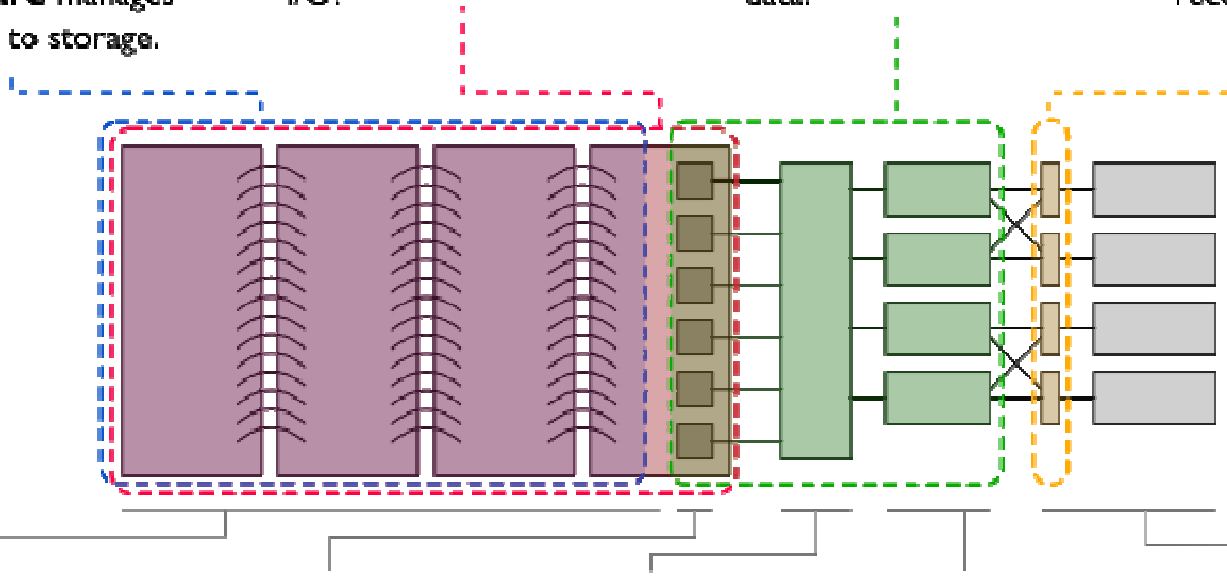
# I/O Hardware and Software on Blue Gene/P

**High-level I/O libraries** execute on compute nodes, mapping application abstractions into flat files, and encoding data in portable formats.
**I/O middleware** manages collective access to storage.

**I/O forwarding** software runs on compute and gateway nodes, bridges networks, and provides aggregation of independent I/O.

**Parallel file system** code runs on gateway and storage nodes, maintains logical storage space and enables efficient access to data.

**Drive management** software or firmware executes on storage controllers, organizes individual drives, detects drive failures, and reconstructs lost data.



**Compute nodes**
*40,960 Quad core PowerPC 450 nodes with 2 Gbytes of RAM each*

**Gateway nodes**
*640 Quad core PowerPC 450 nodes with 2 Gbytes of RAM each*

**Commodity network**
*900+ port 10 Gigabit Ethernet Myricom switch complex*

**Storage nodes**
*136 two dual core Opteron servers with 8 Gbytes of RAM each*

**Enterprise storage**
*17 DataDirect S2A9900 controller pairs with 480 1 Tbyte drives and 8 InfiniBand ports per pair*

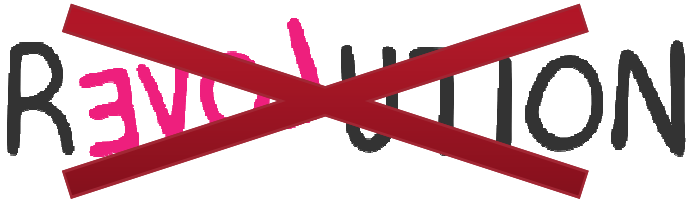
Architectural diagram of the 557 TFlop IBM Blue Gene/P system at the Argonne Leadership Computing Facility.

# Evolutionary or Revolutionary Storage?

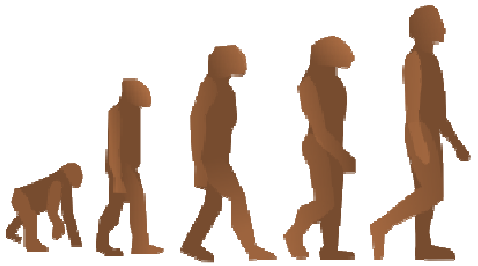
- I like a revolution as much as the next guy, but in HPC I/O, change is slow
  - Unlike companies such as Amazon (S3), Google (bigtable), Facebook (haystack)
  - Working around file system issues is easier than fixing them
  - We need a solution now; New file systems won't be ready for a couple of years…
  - HPC Community is very good at resisting assimilation!

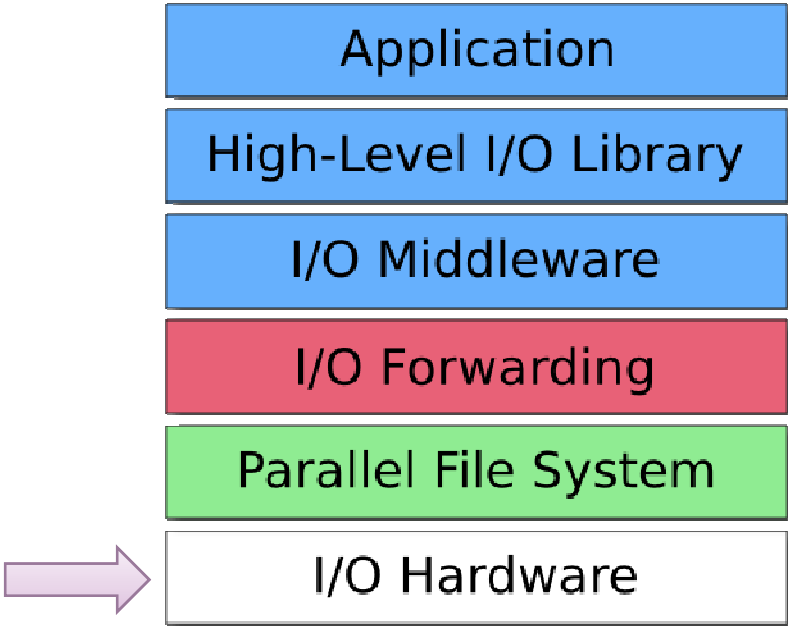
- Can be a bit of both:
  - build high level libraries on top of revolutionary stack
    - MPI-IO (collective open, consistency)
  - Change semantics to what applications require, not what POSIX requires
    - PLFS, Glean,…

- How will parallel file systems look in 10 years?
  - The same!

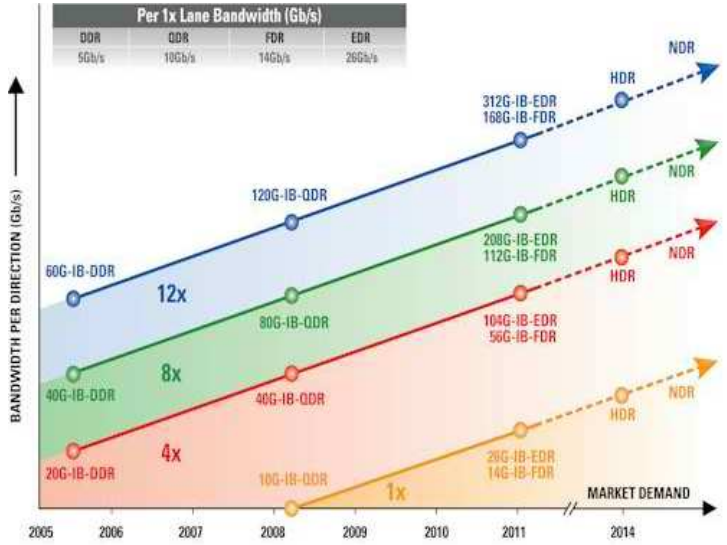


# Evolution: Storage Hardware



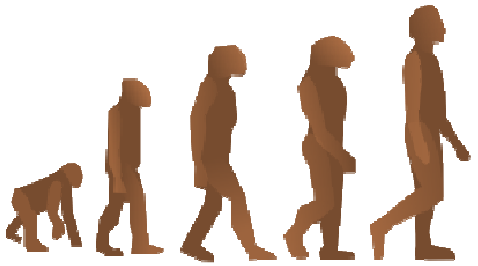


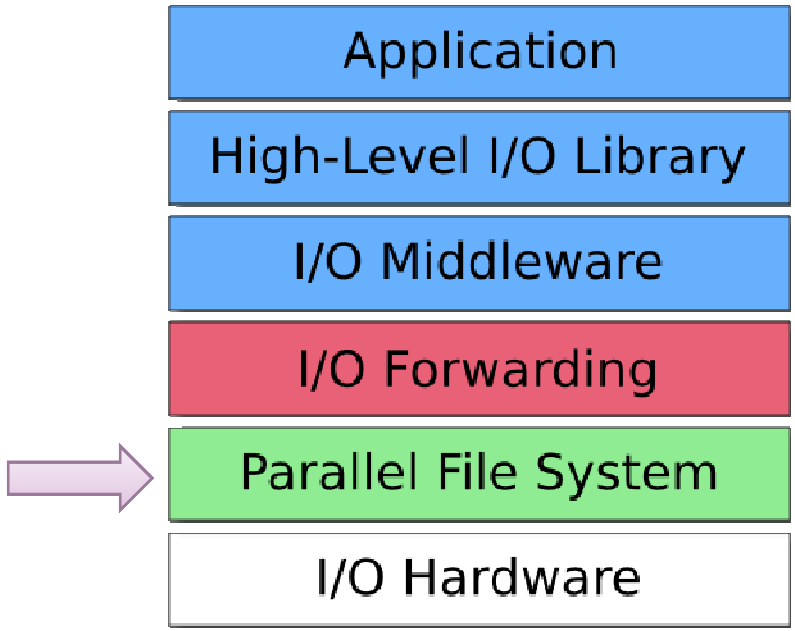

- Storage Media
  - Capacity: Shingled writing
  - Latency: Solid State Drives
  - Phase Change Memory
  - Hybrid drives??
- Network




Source: Infiniband Trade Association

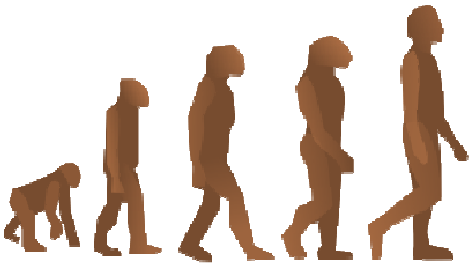
# Evolution: Parallel File Systems

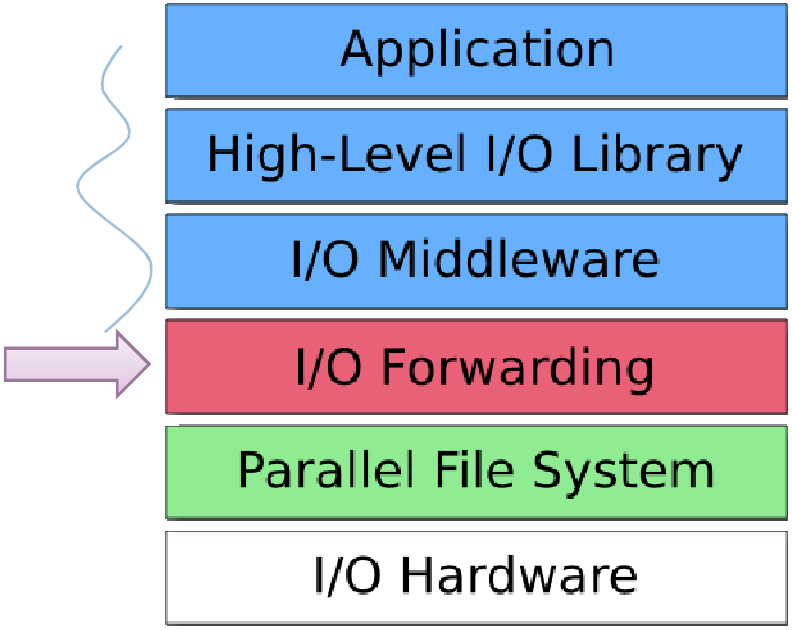





- Faster Metadata
  - Multiple metadata servers
  - Giga+
  - Benefit from Solid State
- Relaxed consistency semantics
  - Lustre group locks
  - Maybe leave it up to middleware?
- Expose data locality
- Don't force the file system to do the dirty work
  - Cfr. Hidden query operations (O_EXCLUSIVE, database-dir, …)

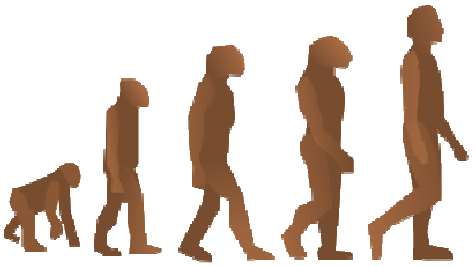
# Evolution: I/O Forwarding

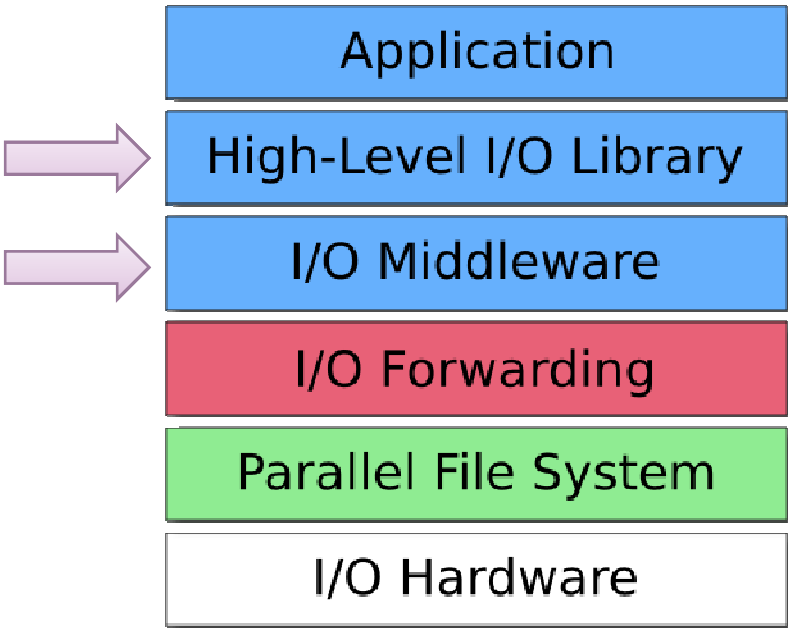





- Distributed Caching
  - Reduces alignment, locking issues
- Node local storage:
  - Data Staging & Write Buffering
  - In-machine scratch (analysis)
- Compression & Deduplication
- New operations
  - Distributed append & return offset
- Resiliency: Fault Tolerant Backplane
- I/O Scheduling
  - QoS
  - Server Directed I/O
- Rework I/O for parallel file system
  - Very performance sensitive
  - Aggregation

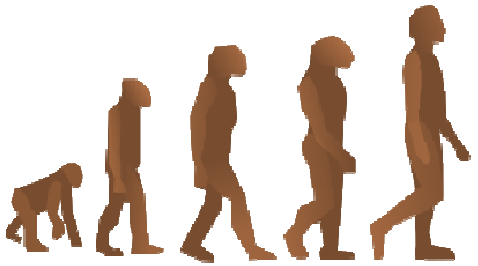
# Evolution

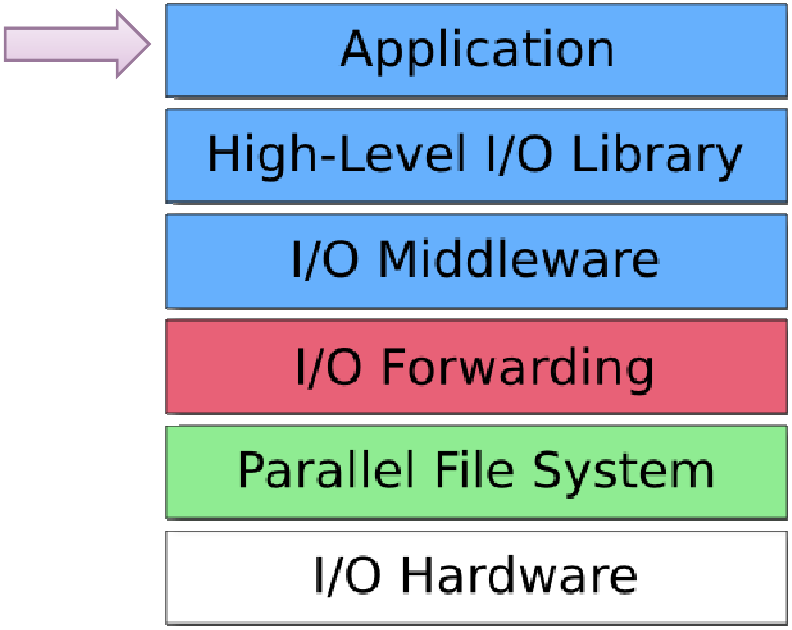


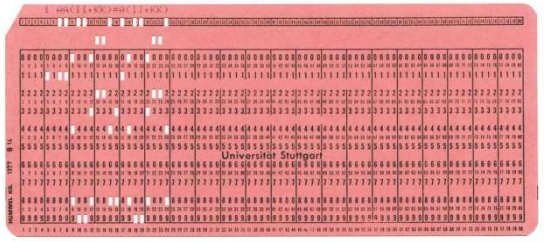



- New domain specific libraries needed
  - Lots of interesting work
- Exploit node local storage?
  - Write buffering, read-ahead
  - Checkpoint draining
- Free cores?
- I/O middleware needs to evolve
  - High level libraries are more complex; Middleware cannot be in the way
  - Interface issues (posix HPC extensions)
- Tuning: automatic or even sysadmin
  - Why don't we have this?
- Data Analysis
  - In situ

# Evolution: Applications






- What? **Modify** my application?
  - Ideally, **yes**.
  - **Maybe** if willing to accept ugly hacks.
- Why?
  - More information
  - Modern **Interfaces, HLL libraries**
  - Rules change (memory, C&C ratio)

# Summary: Evolution, not Revolution

Questions:

- Why do most improvements remain just on paper?
  (or how to speed up **evolution**)
- How to encourage **adoption** of **existing** software and tools?
  - Need to be more user friendly?
- Do we need to bring out our I/O stack as a single monolithic block?
  - Encourage fixing where fixes are needed
  - Make it easier for the user to chose the right HL library
- When will we see Giga+ and PLFS techniques in file systems?
  - Parallel file system evolution is very slow

# Panel Questions

- Will storage be more hierarchical?
  - Yes
- How should we present it to the user?
  - We don't. (note: user = ?)
- Lessons from cloud storage?
  - Different goals
  - If any: see below
- Should we consider record based I/O?
  - Evolution is going that way
- **Crazy idea:** we need an iron hand
  - No longer tolerate misuse/abuse/ignorance
    - But in return support our software
  - Lower layers are root-only

# Acknowledgements


- DOE Office of Advanced Scientific Computing Research (ASCR)
- HEC FSIO Organizers
- Community